

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Tabel 2.1 Tinjauan Pustaka

No	Judul	Teknologi	Hasil
1	Analisis Performa Virtualisasi Container Untuk Penerapan Computer Cluster Sistem Pada Raspberry Pi (Mohammad Hanif, 2018)	Docker, Kubernetes, Raspberry Pi	Mengimplementasikan Virtualisasi Container Dengan Menggunakan Docker Sebagai Container Engine Dan Kubernetes Sebagai Container Management Dan Orchestrator Pada Computer Cluster Menggunakan Perangkat Raspberry Pi Yang Berbasis Arsitektur Arm.
2	Analisis Cluster Container Pada Kubernetes Dengan Infrastruktur Google Cloud Platform (M. Agung Nugroho, Cuk Subianto, 2018)	Docker, Kubernetes	Mengimplementasikan Web Server Berbasis Kontainer Menggunakan Kubernetes
3	Implementasi Failover Dan Autoscaling Kontainer Web Server Nginx Pada Docker Menggunakan Kubernetes (Yosua Tito Sumbogo, Mahendra Data, Reza Andrian Siregar, 2018)	Docker, Kubernetes, Nginx	Mengimplementasikan Web Server Berbasis Kontainer Menggunakan Kubernetes
4	Implementasi Load Balancing Server Basis Data Pada Virtualisasi Berbasis	Docker, Kubernetes	Implementasi Sistem Cluster Container Basis Data Server Pada Virtualisasi

	Kontainer (Moch. Wahyu Imam Santosa, Rakhmadany Primananda, Widihi Yahya, 2018)		Berbasis Kontainer Telah Berhasil Dilakukan Dan Berjalan Dengan Baik Berdasarkan Pengujian Fungsionalitas Replikasi.
5	Jenkins-CI, an Open-Source Continuous Integration System, as a Scientific Data and Image- Processing Platform (Moutsatsos at al., 2019)	Jenkins	Sistem Integrasi Berkelanjutan Open Source, sebagai Data Ilmiah dan Image Pemrosesan Platform
6	Overview of Deployment Options on AWS (Peter Dalbhanjan, 2015)	Amazon Web Service	Gambaran tentang teknologi untuk deployment pada Amazon Web Service (AWS)
7	Implementasi Load Balancing Round Robin .nginx dan Firebase (Restu Ttie Wardhany, Zehan Stanzah, 2017)	Load Balancing, Nginx	Mengimplementasikan Load Balancing dengan metode Round Robin pada nginx dan firebase
8	Amazon Web Services Bootcamp (Sunil Gulabani, 2018)	Amazon Web Services	Implementasi dan cara memulai menggunakan fitur- fitur yang ada di AWS
9	Merancang Aplikasi Perpustakaan Menggunakan SDLC (Dahlan Abdullah , 2017)	System Development Life Cycle (SDLC)	Implementasi Pembuatan Aplikasi Perpustakaan Menggunakan SDLC
10	Rancang Bangun Sistem E- Learning Di SMAN 1 Sekampung Udik Menggunakan Metode Agile (M. Weddy Sumbogo, 2016)	Agile Software Development	Implementasi Pembuatan E- Learning Menggunakan Metode Agile

## 2.2 Dasar Teori

### 2.2.1 Amazon Web Service (AWS)

Merupakan bagian dari perusahaan Amazon yang menyediakan platform on-demand cloud computing bagi individu, perusahaan, maupun pemerintah, dengan sistem berlangganan berbayar. Teknologi yang disediakan oleh AWS yang paling sering digunakan yaitu Simple Storage Service (S3) dan Elastic Compute Cloud (EC2). Untuk mengakses API yang telah disediakan oleh AWS, maka pengguna API membutuhkan pasangan (key-pair) access-key dan secret-key yang dapat di-generate dalam akun pengguna. Access-key dan secret-key merupakan hal yang wajib untuk mencegah terjadinya penyalahgunaan resource dari pihak luar yang tidak mempunyai hak akses terhadap API AWS. Jika pengguna ingin membagi resource yang disediakan kepada pengguna lain namun tak ingin membocorkan informasi mengenai access-key dan secret-key, maka AWS menyediakan IAM (Identity & Access Management). IAM merupakan web service yang memperbolehkan pengguna untuk membuat akun individu bagi orang yang ingin memanfaatkan resource tanpa memerlukan root account. Setiap individu ini akan diberi security credentials yang diperlukan sebagai pembuktian identitas sebagai IAM user. Selain itu root account juga dapat mengatur hak yang dapat diakses oleh tiap IAM user menggunakan group permission, yang dapat diatur sesuai peran dari tiap IAM user misal seperti Administrator, Developer, Accounting, dll.

### 2.2.2 AWS CodeDeploy

AWS CodeDeploy adalah layanan yang mengkoordinasikan penerapan aplikasi di seluruh Amazon EC2. AWS CodeDeploy berfungsi dengan aplikasi yang sudah ada file dan skrip penerapan, dan dapat dengan mudah menggunakan kembali konfigurasi yang ada skrip manajemen. Layanan berskala infrastruktur sehingga dapat menyebarkan ke beberapa EC2 bahkan ribuan. AWS CodeDeploy bisa di gunakan untuk menyebarkan kode program ke infrastruktur yang dikelola sendiri atau organisasi.

### 2.2.3 CI/CD

CI adalah proses pengujian dan pembuatan software secara otomatis setelah kode aplikasi yang baru terintegrasi ke dalam repository (tempat penyimpanan) bersama. Sedangkan CD adalah proses penyampaian aplikasi yang dibuat dalam proses CI ke bagian lingkungan produksi, yang dimasukkan melalui automated test.(Virmani, 2015)

### 2.2.4 AWS CodeBuild

AWS CodeBuild merupakan layanan integrasi berkelanjutan sepenuhnya dikelola untuk mengkompilasi kode program, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap digunakan. dengan CodeBuild tidak perlu lagi mengelola skala server. CodeBuild dapat disesuaikan skalanya secara terus-menerus dan memproses beberapa build secara bersamaan, sehingga build tidak perlu mengantri, ini mempercepat penggunaan environment yang telah di build sebelumnya, dengan CodeBuild juga dapat membuat custom environment sendiri.

#### 2.2.5 AWS CodeCommit

AWS CodeCommit adalah layanan version control yang dikelola oleh AWS dan dapat digunakan untuk menyimpan data secara pribadi (dokumen, file biner, kode sumber) dan mengelolanya di cloud, juga menawarkan skalabilitas tinggi, keamanan dan membantu mengelola layanan kontrol sumber, yang digunakan untuk menghosting repositori Git pribadi. Codecommit mendukung semua fungsi standar yang dapat ditemukan di Git, yang berarti ia bekerja dengan mudah dengan alat berbasis Git pengguna saat ini

#### 2.2.6 AWS CodePipeline

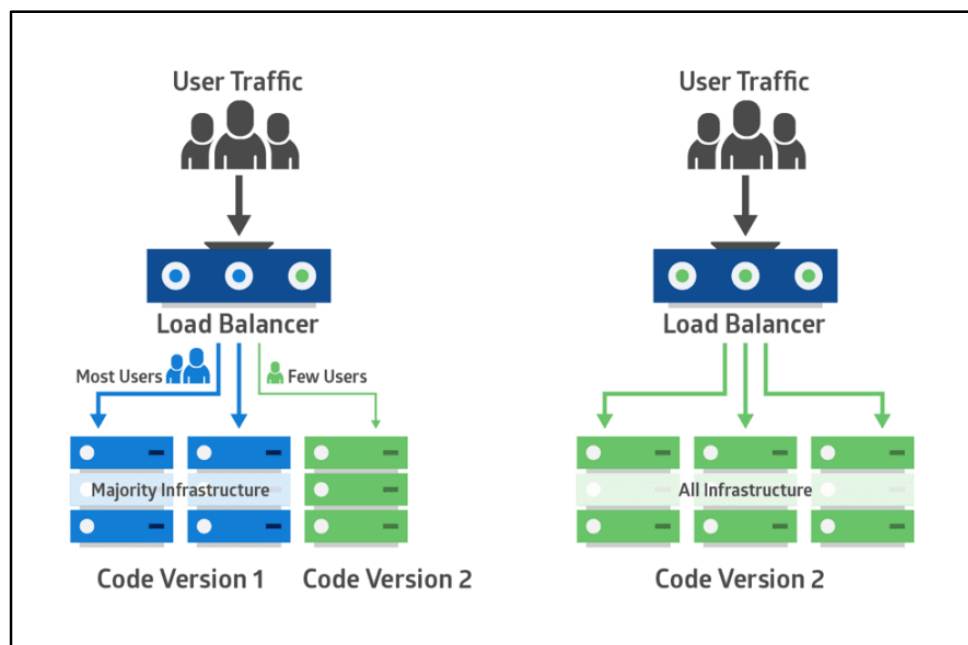
AWS CodePipeline merupakan layanan pengiriman berkelanjutan yang membantu mengotomatiskan alur rilis untuk pembaruan aplikasi dan infrastruktur yang cepat. CodePipeline mengotomatiskan fase build, pengujian, dan deploy proses rilis setiap kali ada perubahan kode, berdasarkan model rilis yang telah ditentukan, Ini memungkinkan untuk mengirim fitur dan pembaruan dengan cepat dan mudah diintegrasikan oleh AWS CodePipeline dengan layanan pihak ketiga seperti GitHub atau dengan plugin kustom sendiri.

#### 2.2.7 Load Balancing

Load balancing merupakan metode pendistribusian kinerja server dalam melayani setiap request yang ada dan membagi permintaan yang masuk untuk diberikan ke suatu server dengan memberikan layanan yang sama

### 2.2.8 Canary Deployment

Canary Deployment adalah dimana penyebaran updatenya dengan cara beberapa user yang dipilih untuk mendapatkan update terbaru dan ini berlangsung sampai semua user mendapatkan update, metode ketiga adalah blue and green deployment hampir mirip dengan canary deployment hanya saja user langsung diarahkan untuk mendapatkan update terbaru tidak secara bertahap.



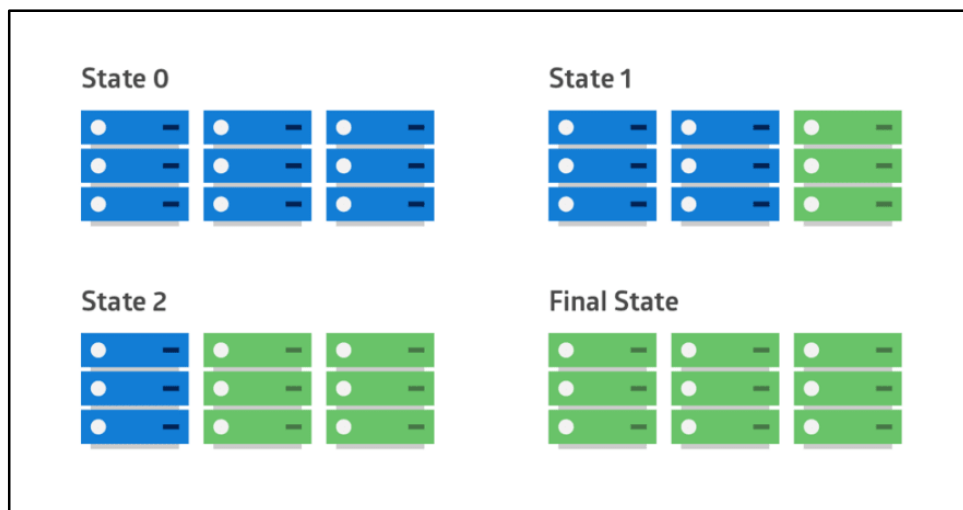
Gambar 2.1 Canary Deployment

### 2.2.9 Rolling Deployment

Penggelaran yang bergulir memperbarui lingkungan besar beberapa node sekaligus. Pengaturan untuk penggelaran bergulir dapat mirip dengan penyebaran biru-hijau, tetapi kali ini dengan lingkungan logis tunggal. Lingkungan dapat memiliki banyak node.

Pengguliran bergulir, bertahap, atau langkah lebih baik daripada penerapan big bang karena mereka meminimalkan banyak risiko terkait, termasuk waktu henti yang dihadapi pengguna tanpa pengembalian mudah.

Dalam penerapan yang bergulir, versi baru aplikasi secara bertahap menggantikan yang lama. Penempatan aktual terjadi selama periode waktu tertentu. Selama waktu itu, versi baru dan lama akan hidup berdampingan tanpa mempengaruhi fungsionalitas atau pengalaman pengguna. Proses ini memudahkan untuk memutar kembali kompo no nonen baru yang tidak kompatibel dengan komponen lama.

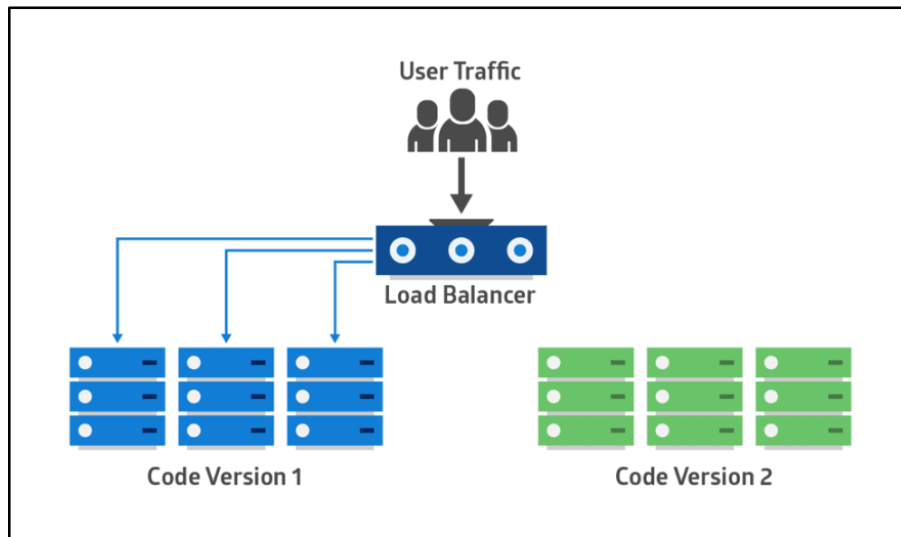


Gambar 2.2 Rolling Deployment

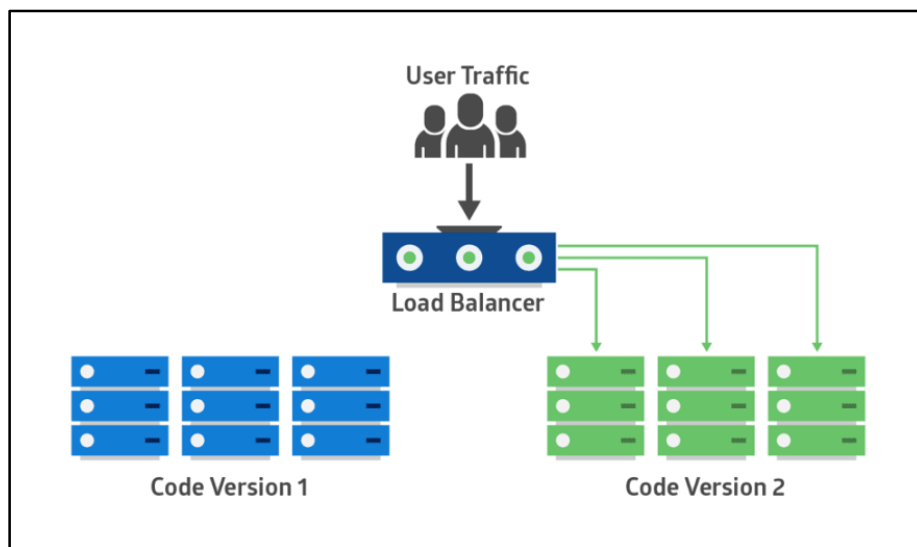
#### 2.2.10 Blue And Green Deployment

Blue/Green Deployment adalah sebuah metode yang di mana proses deployment aplikasi yang dapat mengurangi downtime pada saat proses deployment aplikasi yang telah di update, Dimana ketika pada saat sebuah aplikasi yang telah berjalan akan mendapatkan pembaruan di sebut (green) tidak akan terganggu atau di takedown melainkan akan berjalan seperti seharusnya dan aplikasi yang telah diperbaharui disebut dengan (blue) akan

di deploy dan akan berjalan bersamaan dengan versi sebelumnya.(Sayfan, 2017).



Gambar 2.2.3 Blue Deployment



Gambar 2.2.4 Green Deployment

#### 2.2.10 System Development Life Cycle

System Development Life Cycle (SDLC) merupakan gambaran dari suatu usaha dalam merancang sistem yang akan selalu bergerak seperti roda, yang



melewati beberapa langkah atau tahapan antara lain tahap investigate, analyze, desain, implementasi dan perawatan. Dan langkah selanjutnya akan kembali pada tahap investigate jika di rasakan bahwa sistem yang ada sudah tidak efisien lagi untuk diterapkan. (Dahlan Abdullah, 2017)

#### 2.2.11 Agile Software Development

Agile Software Development adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Agile bersifat cepat, ringan, bebas bergerak, dan waspada. sehingga saat membuat perangkat lunak dengan menggunakan metode Agile Development diperlukan inovasi dan tanggung jawab yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang. ( M. Weddy Sumbogo, 2016)